AAANet: Crowd Counting using Adaptive Scale Aggregation with Attention Networks

Shidong Pan
Group26
u6342277
u6342277@anu.edu.au

Weifan Jiang
Group26
u6683698
u6683698@anu.edu.au

Xiang Gao
Group26
u7096643
u7096643@anu.edu.au

Abstract

Since crowd counting in many real-world scenarios has high application value, it has become a hot topic in academics. Currently, the strategy of using density maps to estimate crowd density for counting is the most popular mainstream method. However, relations between feature map channels are ignored by most of recent works. Hence, we introduce our model: Adaptive Scale Aggregation with Attention Networks (AAANet). Basic block of AAANet contains 2 main parts: Aggregation module which extracts features and attention module to integrate the incoming features. It combines the benefit of residual learning and attention mechanism. And the model is tested and obtain a decent results on ShanghaiTech dataset Part A: 78.3 in MAE and 106.5 in MSE; Part B: 21.4 in MAE and 32.8 in MSE.

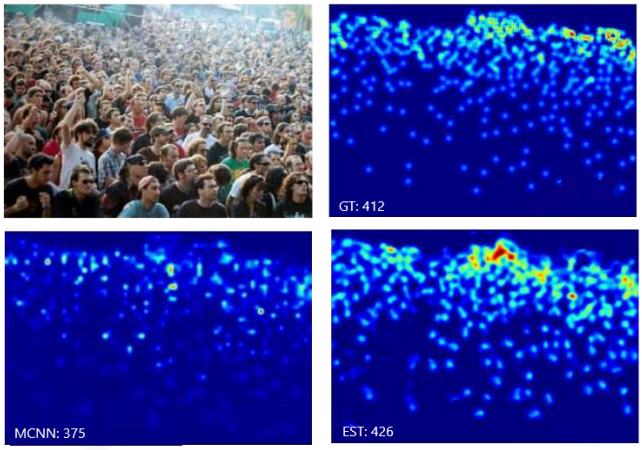

Figure 1. Four figures above from top left to bottom right are the real image in ShanghaiTech Part B dataset, groud truth, result of MCNN and ours respectively. Numbers of individuals are marked

1. Introduction

Due to the COVID-19 pandemic, keeping social distancing in public becomes a common sense and an effective method to encounter the spread. In practice, the government limits and regulates the maximum number of people in a public area to guarantee the social distancing. Thus, crowd counting has become an increasingly important application for crowd size controlling and public safety. However, crowd counting remains an open problem because of the special properties of the crowds: Pedestrians may be clustered, overlapped and occluded, and they can have different sizes and shapes from different perspectives. To study this problem, various approaches and algorithms are developed and enjoys a wide spectrum of applications in reality.

1.1. Review of standard algorithms

Classical algorithms to address crowd counting are mainly divided into three paradigms: Based on Detection, Regression, and Density estimation.

In the first beginning of this field, researchers focused on developing a model that can detect human bodies from the input images. The detector uses a sliding window to filter out the human features to count the number [3]. Detection performs three different types according to the features it detects: monolithic style, part-based detection and shape matching. For monolithic detection, it trains a classifier that extracts features such as Haar wavelets [11], histogram rriented gradients(HOG) [10] and edgelet part [20] from full human appearance [2, 23]. Monolithic detection has been successfully implemented in many models, but it only works in sparse crowds counting and receives unsatisfactory performance in high-density crowd scenes. In a crowded scene, pedestrians are obscured, and it is hard to detect a monolithic target. In order to solve this issue, researchers developed an approach that transforms detection from whole body to part of local bodies. This method tracks body parts like head or shoulders to recognize [21]. Another algorithm is to use shape matching.

Zhao *et al*. [27] used ellipsoids to build human shapes in 3D and matched the best solution in a stochastic process. In general, the detection algorithm for crowd counting is relatively expensive and complex, and does not perform well in high-density situation.

To overcome the drawbacks of detection methods in crowed pedestrians images, a new approach that maps features extracted from local image patches to the number of crowds is developed and gains a better performance on the problem of high-density population counting [9]. In a standard regression model, it first extracts low resolution from the image and then does the regression modelling. Features like edges, gradients, foreground or textures can be used as low-level information. One obvious advantage of this method is that it can get a population count without detecting individuals.

In 2016, Xu *et al*. proposed to use rich features and random projection forest as regression models. This improved performance while avoiding the problems caused by high dimensionality, made their models perform better than all of the models in the [22]. In 2020, Mixture regression framework with three modules, 3D Gaussian regression with multi-view fusion, bi-Branch attention network and many other models are developed to give more real-time and precise results in crowd counting [12, 7, 25].

Although the former two methods can complete the task of crowd counting to some extent, they cannot describe the spatial information between features which is quite significant in crowd estimation topic. Density estimation can handle this problem by learning the mapping between local patch features and crowds density map [9]. This is also the primary method used in this study. In this way, the density estimation not only avoids the complex detection task, but also avoid to carry out regression learning for different feature images, respectively. It directly maps the population distribution and spatial relationship by establishing a density-map and counts the number by the integral method. It means the crowd counting problem can be formulated as an optimization task of a regularized risk quadratic cost function which is intuitive and straightforward structured.

Pham *et al*. [13] are the early researchers who developed a nonlinear density estimation model applied to random decision forests. In addition, they raised a crowdedness prior and a forest reduction method to improve the speed and precision of the counting model. In the following other density estimation models, Two-Column Convolutional Neural Network (TCCNN) which derives from VGG-16 and Alexnet is implemented to give a better performance [14], and some mixture models like CNN-RNN are transferred into crowd counting tasks. For example, in 2018, Fu *et al*. built a model called CRCCNN, which combined recurrent neural network with standard convolutional neural network and used Long Short-Term Memory (LSTM) structure to remember the input information of sequential samples [4]. This gives the model better performance when extracts the contextual information of crowd region and estimates precisely with high-density crowds.

1.2. Our work

The number of pedestrians in a scenario can be calculated based on either multiple images from different perspectives or a single image from a fixed angle. The previous situation can be solved by 3-D scenario reconstruction methods, and in practice, crowd counting on a single image is more common and has a wider application potential. In this report, when we discuss crowd counting, it indicates the single image scenario.

To keep improving the performance on methods based on density map, firstly, we briefly analysis and review several previous famous milestone models. Furthermore, many previous works showed that attention mechanism exhibits a remarkable performance on other deep learning tasks. We notice that current discussion focuses on how to extend the receptive filed by novel methods, but neglects the importance of appropriately integrating them. Thus, we use squeeze-and-excitation block to implement attention mechanism and combine with a classic network: ResNeXt to explore a potential improvement. To summarize, the following are our main contributions:

- We propose a novel Adaptive Scale Aggregation with Attention Networks (AAANet) for crowd counting and density estimation.
- To the best of our knowledge, ours is the first attempt to embed SE blocks to implement attention mechanism on crowd counting tasks to better aggregate feature maps under different receptive fields.
- Extensive experiments are conducted on a highly challenging dataset ShanghaiTech [26], and comparisons are performed against several recent state-of-the-art approaches.

2. Related work

2.1. SENet

Squeeze-and-excitation network(SENet) [8] was proposed by Hu *et al*. in 2018, and a SENet is composed of multiple SE blocks stacking together. Expressly, a SE block mainly undertakes two tasks: squeeze and excitation. In squeeze step, after the global average pooling, the

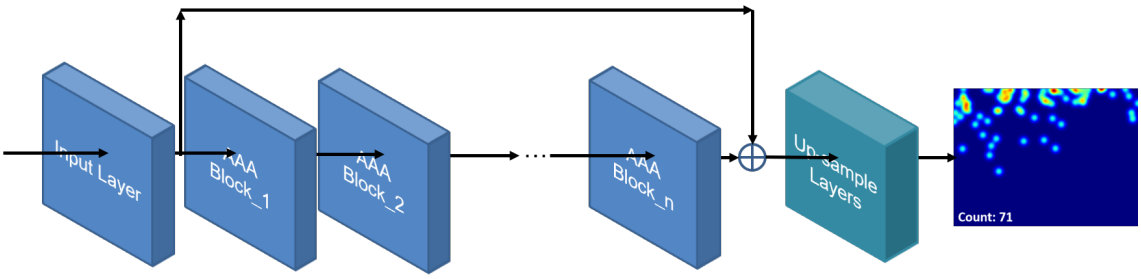

Figure 2. The framework of proposed Adaptive scale Aggregation with Attention model for crowd density map estimation. The pipeline of our model first comes with an input layer, then multiple AAA blocks with deep residual learning, and up-sample layers at last to get the density map.

coarse global information can be extracted and converting from $H \times W \times C$ into a shape of $1 \times 1 \times C$. As for the excitation step, to gain the nonlinear interaction and non-mutually-exclusive relationship between channels, a simple gating mechanism is employed by authors with a Sigmoid activation function, followed by rescaling. The SE block has the same functionality as a self-attention mechanism. In practice, those novel blocks can be embedded into other networks to gain better performance by increasing the learning ability of the networks on latent relationship and patterns. In terms of computational complexity, it dramatically decreases the amount of parameter and saves much computation resource, improving efficiency as well.

2.2. ResNeXt

ResNeXt is a network model that combines two ResNet architecture [6] and Inception module [17]. Compared with ResNet, we see that ResNeXt extents the single convolution into group convolution with fewer channels, meanwhile the signature shortcut in ResNet keeps unchanged. As for the improvement in respect to inception module, ResNeXt normalizes the inception unit, makes the convolutional structure not need delicate design and organized parameters. By setting the cardinality as a parameter, ResNeXt can easily achieve any possible structure. Furthermore, the authors point out that split-transform-merge pattern is a standard pattern of general neural network structure, and ResNeXt is a simple implementation of this pattern. Therefore, all those advantages of this model inspire us to explore more possibility of it with other structures on the crowd counting field.

2.3. MCNN

Multi-column Convolutional Neural Network (MCNN) was introduced by Zhang *et al*. [26] on CVPR 2016, which aims to overcome the poor performance of traditional crowd counting methods on the dataset with a large variety of crowd density and distribution. Specifically, by applying multiple convolution columns with different kernel sizes to replace tradition single column convolution, the receptive field of the model extends. Thus it can adapt different people figure size in the input image. Also, using convolution layer with 1×1 kernels instead of full connection layer enables the model can receive input in any size, avoid deformation and directly gain the final image. Therefore, by applying multi-column structure, effective features in different sizes are automatically extracted and integrate together. In addition, the ShanghaiTech dataset is also collected and published with this paper

2.4. CP-CNN

.

Contextual Pyramid CNN (CP-CNN) [16] is a following work of MCNN, and by explicitly incorporating global and local contextual information of crowd images, CP-CNN gains better results than MCNN. Firstly, Global Context Estimator (GCE) and Local Context Estimator (LCE) are two networks based on VGG-16, parsing global or local context information respectively to divide input images into different density levels. Then, those contextual information fuses with high-dimensional feature maps from a multi-column architecture-based CNN (DME), which is similar to the MCNN structure. Finally, F-CNN is trained along with the DME in an end-to-end fashion to generate high high-quality density maps.

2.5. SANet

Similar to previous two methods, Scale Aggregation Network (SANet) [1] also concentrates on extracting multi-scale features with scale aggregation modules. Moreover, authors broaden the receptive field by introducing the new structure that looks like Inception module [17] instead of multi-column design in MCNN. Briefly speaking, different size convolution kernels are used at each convolution layer simultaneously to extract features at different scales, and finally, the final density map is obtained by transposed convolution to generate the density map at high resolution.

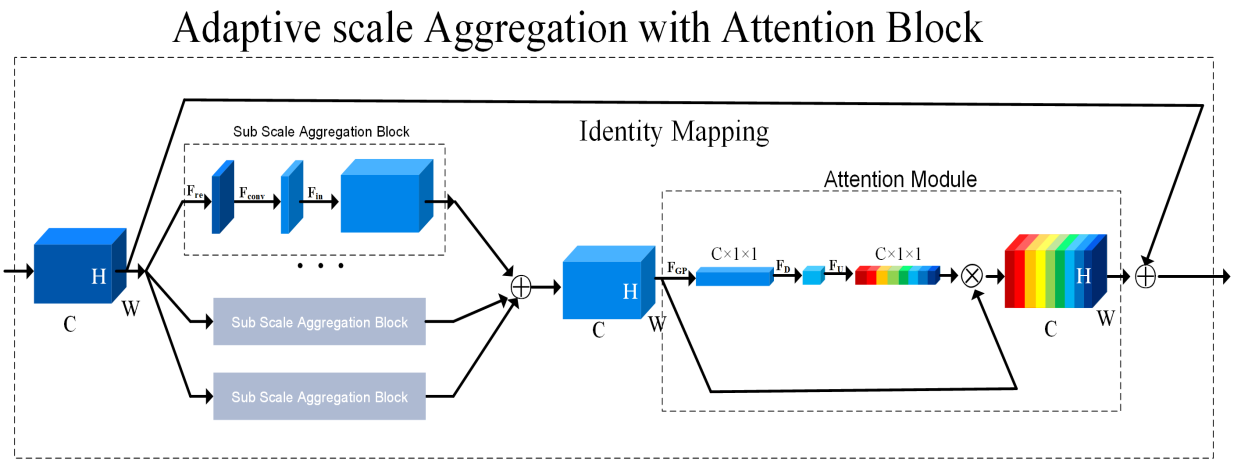

Figure 3. The figure presents the architecture of an Adaptive scale Aggregation with Attention Block. In a standard block, the input is first assigned to two branches. (Top right)The first branch feeds the input to the final output with identity mapping using a shortcut connection. Another branch leads to the scale aggregation network. A subscale aggregation block consists of a dimension reduction function F_{re}, a standard convolutional function F_{conv} and a dimension recovery function F_{in}. Scale aggregation feeds inputs to multiple sub-modules with the different kernel size of the function F_{conv} and integrates their outputs into one feature graph. Then we commit the result into Attention module. Through F_{GP} (Global pooling), F_D (FC+ReLU), F_U (FC+Sigmoid) and weight concatenation, our model achieves attention capacity on specific features. At last, the weighted feature map will be combined with the residual part and output.

3. Proposed method: AAAN

3.1. Network pipeline

As shown in Figure 2, we adopt module design for the network architecture. The entire network contains three parts: input layer for patch extraction, A network body composed of several Adaptive scale Aggregation Attention Blocks (AAAB) for non-linear mapping, and the final output layer to generate the final density-map.

In terms of the pipeline, one convolution layer is adopted to extract the patch from the input:

$$Patch = F_0(I), \tag{1}$$

where $F_{input}(\cdot)$ is the convolution operation, and I presents input image. Then $Patch$ is sent to body network with AAAB module to obtain the feature map. therefore, we have:

$$Features = F_{AAAB\#}(Patch), \tag{2}$$

where $F_{AAAB\#}(\cdot)$ denoted as body network with AAAB module, and is the number of the AAAB modules of the body network. At last stage, the up-sampling layer fits the Feature map to target size :

$$Feature_{up} = F_{up}(Feature_{map}), \tag{3}$$

where $F_{up}(\cdot)$ is the up-sampling operation, that has several options like transposed convolution Nearest-Neighbor with one convolution, and etc. From the problem definition 1.2, we can see the similarity between crowd counting tasks and segmentation tasks: Both of them leans a mapping pixel-to-pixel. The formal one predicts a Density values for every pixel, and the latter classifies every pixel. Hence, we used a transposed convolution for size fitting, and then a convolution operation, denoted as F_g, to generate the density map:

$$D_{map} = F_g(Feature_{up}), \tag{4}$$

After getting the output, Mean Square Error (MSE) loss is used for optimizing the network:

$$L(\Theta) = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left[GT(i,j) - D_{map}(i,j) \right]^2, \tag{5}$$

where Θ is the parameters of the proposed model, m,n presents the width and height of the density-map. Therefore, the goal of the pipeline is to minimize the $L(\Theta)$ loss function and get our proposed model: After getting the density-map, we can sum all the output pixels to get the statistics of the number of people in the corresponding scene:

$$counts = sum(D_{map}) \tag{6}$$

3.2. AAAB

Details of the AAAB module structure is clearly illustrated in Figure 3. We see that AAAB contains two main conponents, Aggregation module and attention module. For a H-by-W-by-D input at i-th AAAB, denoted as $I_i = [I_{i1}, I_{i2}, ..., I_{iD}]$, First, it will be sent into 4 paths. After feature extraction in different path, feature map will

Table 1. Estimation errors on the ShanghaiTech dataset.

	Part A		Part B	
Method	MAE	MSE	MAE	MSE
Zhang *et al.* [24]	181.8	277.7	32.0	49.8
MCNN [26]	110.2	173.2	26.4	41.3
Switching-CNN [15]	90.4	135.0	21.6	33.4
SANet [1]	90.4	135.0	21.6	33.4
CP-CNN [16]	**73.6**	**106.4**	**20.1**	**30.1**
ours	78.3	116.5	21.4	32.8

be concatenated together and reformed to original dimension, denoted as $O_i = [O_{i1}, O_{i2}, ..., O_{iD}]$. Then it will be passed to our attention module. The attention factor s between channels of O_i can be generated by down-sampling, i.e. shrink the height and width of O_i to one by a average pooling:

$$s_{id} = avgpooling(\boldsymbol{x}) = \sum_{i=1}^{H}\sum_{j=1}^{W} O_{id}(i,j), \tag{7}$$

where s_{id} is the attention factor in i-th AAAB d-th dimension, $d \in D$. Next, we set two convolution layers and two activation functions to grab non-linear relation between each dimension:

$$z_i = sigmoid(W_{i1} * (ReLU(W_{i2} * s_i))), \tag{8}$$

where W_{i2} and W_{i1} is the weight of two convolution layers, $*$ presents convolution operation. At next stage in AAAB, the final attention factor z_i will summarize the feature map O_i:

$$\hat{O}_i = O_i \cdot z_i, \tag{9}$$

where $\cdot$ is element wise product. At final stage of the AAAB, a short cut is added:

$$\hat{I}_i = \hat{O}_i + I_i, \tag{10}$$

, where $\hat{I}_i$ is final output of i-th AAAB and input of $(i+1)$-th AAAB.

3.3. Density-map Generation

Compared with an integer, the density map can contain more information, such as a spatial distribution of the crowd. To obtain the Ground Truth as required, Zhang *et al.* [26] proposed KNN density map algorithm with delta function: $\delta(x - x_i)$. Given a image with N head, for each head x_i, $\{d_1^i, d_2^i, d_3^i, ..., d_m^i\}$, and the mean value of these distances is $\bar{d^i} = \frac{1}{m}\sum_{j=1}^{m} d_j^i$. The density is:

$$H(\boldsymbol{x}) = \sum_{i=1}^{N} \delta\left(\boldsymbol{x} - \boldsymbol{x}_i\right) * G_\sigma, \sigma_i = \alpha \bar{d^i} \tag{11}$$

where $*$ means convolution operation, G_σ is Gaussian kernel with variance σ_i proportional to $\bar{d^i}$, in practice the value of the β usually set to 0.3. Equation 11, in another words, generates a Gaussian kernel for every crowd scenes and convolve with it. The Figure 1 shows the original image and Ground Truth pair.

4. Experiments

In this section, we will introduce our parameters, training settings, and demonstrate results.

4.1. Training details

Datasets and models ShanghaiTech is a large-scale crowd counting dataset including 1198 images. It contains two parts: Part A and Part B. Part A is composed of 482 images, which are randomly selected from the Internet, and Part B is from the Shanghai urban area on the street. We set the number of the AAAB to 200 and trained two models under Part A and Part B.

Evaluation metrics The density-map are evaluated with Mean Absolute Error (MAE) and Mean Square Error (MSE). Besides, we show comparison with different state-of-the-art models.

Training settings Due to the very deep network structure, we cut the image into a 96-by-96 mini-patch for training. Random horizontal flipping and random rotation operation are added as well, to enhance robust of our model. The learning rate was set to 0.0001 and training for 600 epochs with Adam optimizer. the hardware and software platform shows at Table 2.

4.2. Results

Table 1 shows the quantitative statistic of our model compared with other models. Our model is compared with five approaches: Zhang [24], MCNN [26], Switching-CNN [15], SANet [1], and CP-CNN [16] on ShanghaiTech. Those technique detail of the methods, we discussed in 1.1. It can be seen from the table that our network has better performance than any other methods except state-of-the-art CP-CNN.

From Figure 4, it can be seen that the positions of people's heads are well marked, whether crowd is evenly dis-

Table 2. Experiment environment settings.

		specs.
Software	System	Windows 10 professor 64-bit
	Python	3.7.4
	Torch	1.7.0
	CUDA	10.2
Hardware	CPU	Intel(R) Core(TM) CPU i5-9600K @ 3.70GHZ
	GPU	Nvidia GeForce RTX 2070 (8GB)
	Memory	16GB DDR4 2400MHz

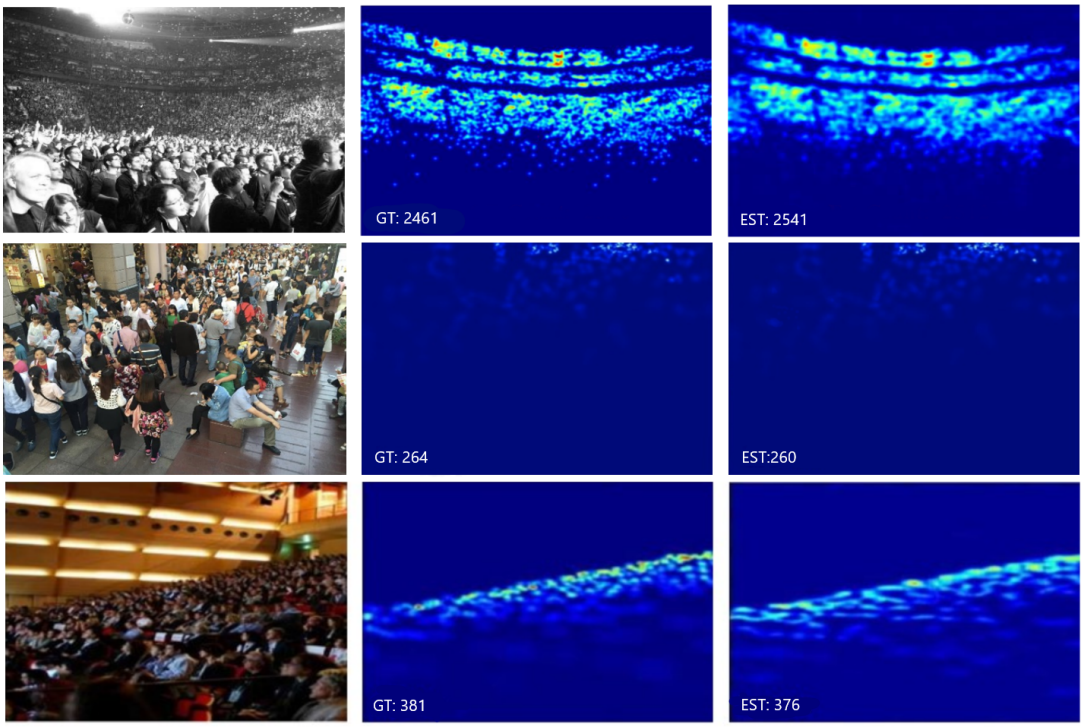

Figure 4. Original image, ground truth density map and estimated density map of our AAANet Model of Three test images in part A.

tributed on the street or in an extreme crowd stadium.

5. Conclusion and Discussion

We proposed a new architecture on crowd counting tasks, and see a good improvement. The aggregation module in the boy network can grab feature map with different perception fields. The deployment of attention mechanism enables network leans relations between feature map in different convolution layers. From all the above, this paper is a meaningful, positive attempt in crowd counting research filed. With the proposed network, MAE and MSE are reduced to 78.3 and 116.5 for ShanghaiTech Part A, 21.4 and 32.8 for Part B. Although we did not beat the state-of-art performance, the experiments prove that attention mechanism does contribute to gain a better integration for feature maps.

In terms of the feature work, first, like CP-CNN [16], we may try a network with three parallel boy network for density-features, global features and local features. Besides, the advantages of the architecture is that the attention factor could be determined by global or local features with the same layer. Hence, we expect combining of the all-around features can bring improvement in our model.

Second, combining different loss functions is worthy of trying as well. For example, the counting number be soothed into 100 classes, each bin between classes set to 20. Then we could deploy cross-entropy loss. With the same inspiration, We could add Center loss [19] or Contrastive loss [5]. This type of loss strategy has already seen in representation learning, such as ReID [18]. By doing this, our model would recognize the sparseness of people in the scene, which generalize our model.

References

[1] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. Scale aggregation network for accurate and efficient crowd counting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018. 3, 5

[2] Cheng-Hsiung Chuang, Shih-Shinh Huang, Li-Chen Fu, and Pei-Yung Hsiao. Monocular multi-human detection using augmented histograms of oriented gradients. In *2008 19th International Conference on Pattern Recognition*, pages 1–4, 2008. 1

[3] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):743–761, 2011. 1

[4] Jingnan Fu, Hongbo Yang, Ping Liu, and Yuzhen Hu. A cnn-rnn neural network join long short-term memory for crowd counting and density estimation. In *2018 IEEE International Conference on Advanced Manufacturing (ICAM)*, pages 471–474. IEEE, 2018. 2

[5] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 6

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[7] Yi Hou, Chengyang Li, Fan Yang, Cong Ma, Liping Zhu, Yuan Li, Huizhu Jia, and Xiaodong Xie. Bba-net: A bi-branch attention network for crowd counting. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4072–4076. IEEE, 2020. 2

[8] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 2

[9] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In *Advances in neural information processing systems*, pages 1324–1332, 2010. 2

[10] Guoyun Lian. Pedestrian detection using quaternion histograms of oriented gradients. In *2020 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS)*, pages 415–419. IEEE, 2020. 1

[11] Sheng-Fuu Lin, Jaw-Yeh Chen, and Hung-Xin Chao. Estimation of number of people in crowded scenes using perspective transformation. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 31(6):645–654, 2001. 1

[12] Xiyang Liu, Jie Yang, and Wenrui Ding. Adaptive Mixture Regression Network with Local Counting Map for Crowd Counting. *arXiv e-prints*, page arXiv:2005.05776, May 2020. 2

[13] Viet Quoc Pham, Tatsuo Kozakaya, Osamu Yamaguchi, and Ryuzo Okada. Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015. 2

[14] Jianing Qiu, Wanggen Wan, Haiyan Yao, and Kang Han. Crowd counting and density estimation via two-column convolutional neural network. 2017. 2

[15] Deepak Babu Sam, Shiv Surya, and R Venkatesh Babu. Switching convolutional neural network for crowd counting. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4031–4039. IEEE, 2017. 5

[16] Vishwanath A Sindagi and Vishal M Patel. Generating high-quality crowd density maps using contextual pyramid cnns. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1861–1870, 2017. 3, 5, 6

[17] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 3

[18] Rahul Rama Varior, Mrinal Haloi, and Gang Wang. Gated siamese convolutional neural network architecture for human re-identification. In *European conference on computer vision*, pages 791–808. Springer, 2016. 6

[19] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016. 6

[20] Bo Wu and Ramakant Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 90–97. IEEE, 2005. 1

[21] Bo Wu and Ram Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2):247–266, 2007. 1

[22] Bolei Xu and Guoping Qiu. Crowd density estimation based on rich features and random projection forest. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8. IEEE, 2016. 2

[23] Junjie Yan, Zhen Lei, Dong Yi, and Stan Z Li. Multi-pedestrian detection in crowded scenes: A global view. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3124–3129. IEEE, 2012. 1

[24] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. Cross-scene crowd counting via deep convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 5

[25] Qi Zhang and Antoni B. Chan. 3D Crowd Counting via Multi-View Fusion with 3D Gaussian Kernels. *arXiv e-prints*, page arXiv:2003.08162, Mar. 2020. 2

[26] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 589–597, 2016. 2, 3, 5

[27] Tao Zhao, Ram Nevatia, and Bo Wu. Segmentation and tracking of multiple humans in crowded environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(7):1198–1211, 2008. 2